

APPENDIX B1

HYDRODYNAMIC MODEL TRAINING MANUAL

**OTB INTEGRATED MODEL
SYSTEM
HYDRODYNAMIC MODEL
MODEL CALIBRATION MANUAL**



0 1 2 Miles

INTRODUCTION

This training manual has been prepared to support the calibration components for the Hydrodynamic Model of the OTB Integrated Model System. The objectives of this training manual is as follows;

- Provide instruction to users on the HYDRODYNAMIC MODEL structure and how to set-up and run the model for the OTB Integrated Model System
- Provide tools/instruction to allow users to produce graphics and statistics using the model calibration graphics and statistics as an example

Based upon these objectives, this manual provides specific descriptions and instructions for the OTB specific Hydrodynamic Model application, and is not meant as a general user's manual. The model manual has been provided with the model files and provides more general instruction on the capabilities and options available. Additionally, this manual provides instructions on how to post-process the output to generate the graphics and statistics from the model calibration report. Instruction on post-processing for assessment of alternate model scenarios will be provided in future training sessions.

The instructions are presented in three primary steps, these are;

- Step 1: Setting up the Hydrodynamic Model control and input files
- Step 2: Running the Hydrodynamic Model
- Step 3: Hydrodynamic Model post-processing for calibration graphics and statistics

STEP 1: SETTING UP THE HYDRODYNAMIC MODEL CONTROL AND INPUT FILES

In order to run the Hydrodynamic Model application for the Integrated OTB model system, four files are needed. These are;

- **run_data**: the primary input file for the Hydrodynamic Model; includes the model run control, model coefficients, model output control, and boundary conditions
- **model_grid**: contains the grid geometry data, the depths, the specification of the vertical layers, and the specification of the thin wall dams
- **groundwater.map**: provides information on zoneation of the model grid for the purpose of defining groundwater inflows spatially
- **ecom3d.exe**: the executable file for the Hydrodynamic Model (for this version)

The time dependant boundary conditions included in the **run_data** file are generated from time series data stored within ACCESS databases provided with the model input files. Figure 1 below provides a schematic representation of the ACCESS database structure, the specific data sets that feed into the 6 ACCESS database files that contain the boundary condition information, and the flow of data from the 6 ACCESS databases to the **run_data** file.

The specific ACCESS database files which contain the data for each of the boundary conditions listed above are provided within the blue circles in Figure 1. Additionally, the specific data input (blue box) and the individual file names for each input (green boxes) are shown. For each of the six boundary conditions a single ACCESS database file is created which merges all of the

data within the circle. The final output files are shown attached to the circle feeding into the **run_data** file. The 6 ACCESS files are as follows;

- **ELEVATION.accdb**: Time series of water levels applied uniformly to the Hydrodynamic Model offshore boundary.
- **SALTEMP.accdb**: Time series of salinity and temperature applied uniformly to the OTB hydrodynamic model grid offshore cells.
- **RIVERFLOW.accdb**: Time series of tributary inflows and temperatures input at specific model grid locations.
- **DIRECTPS.accdb**: Time series of direct point source inflows and temperatures input at specific model grid locations
- **GROUNDWATER.accdb**: Constant groundwater inflows and temperatures applied throughout the OTB hydrodynamic model
- **MET.accdb**: Time series of meteorologic data applied throughout the OTB hydrodynamic model

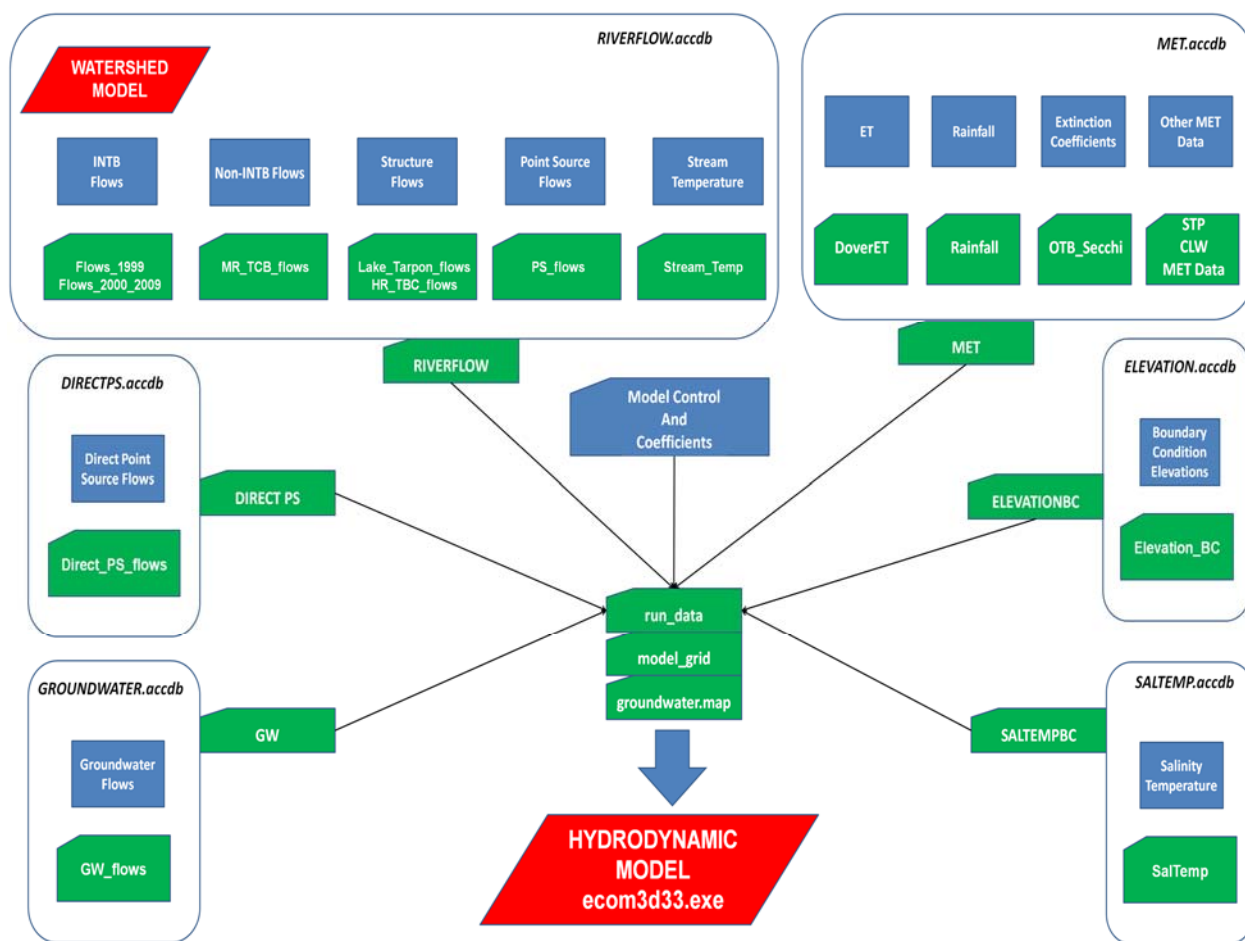


Figure 1: Hydrodynamic Model Input File Structure

The following provides detailed descriptions of all of the variables within the three model input files. Additionally, detailed instructions are provided for creation of the boundary conditions in the **run_data** file from the ACCESS databases.

run_data

The **run_data** file is broken down into 8 separate Data Groups. Appendix A presents a detailed discussion of the variables, options, and input formats for each of the Data Groups. The specific Data Groups include;

- Data Group A – Computational and Diagnostic Characteristics
- Data Group B – Hydrodynamic Characteristics (coefficients)
- Data Group C – Result Evaluation (Output)
- Data Group D – Standard Level Declaration
- Data Group E – Initial Temperature and Salinity
- Data Group F – Open Boundary Conditions
- Data Group G – Inflow Boundary Conditions
- Data Group H – Meteorologic Boundary Conditions

Under Data Group F, two sub-groups of open boundary forcing data exist, these are

- Sub-data Group F1 - water surface elevation data
- Sub-data Group F2 - salinity and temperature data

Under Data Group G, three sub-groups exists, these are;

- Sub-data Group G1 – tributary inflows and temperatures
- Sub-data Group G2 – direct point source discharges and temperatures
- Sub-data Group G3 – groundwater inflows and temperatures by zone (based on **groundwater.map** file)

The following outlines the steps for creation of each of the sub-groups. The USER is referred to Appendix A for details on the specific formats for each group.

Step 1-1: Define the model controls and the type of model to run using the variables in data group A. This includes;

1. Run options
2. Run computational characteristics (time step, etc)
3. Run output characteristics
4. Run print characteristics

Step 1-2: Define the model coefficients using the variables in data group B. This includes;

1. Model coefficients
2. Horizontal mixing coefficients
3. Vertical mixing coefficients

Step 1-3: Define the model output using the variables in data group C. This includes;

1. Model output control for GCMPLT file
2. Output interval for time series in GCMTSR file
3. Model grids for elevation time series into GCMTSR
4. Model grids for velocity, salinity and temperature time series in GCMTSR
5. Model grids and directions for flux output to GCMTSR

6. Model output control for GCM_TRAN file to water quality model

Step 1-4: Define the vertical depth levels for calculation of the density gradients that feed into the Hydrodynamic Model (Standard Level Declaration) for data group D.

1. Number of vertical layers
2. Fixed depths for each layer

Step 1-5: Define the fixed salinity and temperature initial conditions by Standard Level for data group E.

1. Input fixed initial temperature for each layer
2. Input fixed initial salinity for each layer

Step 1-6: Input the water surface elevation forcing function for data group F1

1. Define the number of cells over which the water surface elevation forcing is applied
2. Define the grids and the connecting grids (I,J) for the water surface elevation forcing
3. Input the time series data for each defined grid as block data using the ACCESS database "ELEVATION.accdb"
 - a. This database contains one primary data table, "Elevation_BC", which includes time and elevation data at 6-minute intervals
 - b. Export four BC files
 - i. ELEVATIONBC1.xls" - Hour 0 - 24999.976
 - ii. ELEVATIONBC2.xls" - Hour 25000.076 - 49999.976
 - iii. ELEVATIONBC3.xls" - Hour 50000.076 - 74999.976
 - iv. ELEVATIONBC4.xls" - Hour 75000.076 - 96430.176
 - c. Remove header row (Row 1) from each file
 - d. Set format to 5 decimals
 - e. Set column width to 11
 - f. Save as: "ELEVATIONBC1.prn", "ELEVATIONBC2.prn", "ELEVATIONBC3.prn", "ELEVATIONBC4.prn"
 - g. Run executable: "elevationbc.exe"
 - h. Creates formatted file for putting into "run_data": "ELEVATIONBC"

Step 1-7: Input the salinity and temperature forcing function for data group F2

1. Input the time series data for each defined grid as block data using the ACCESS database "SALTEMP.accdb"
 - a. This database contains one primary data table, "SalTemp", which includes monthly water column EPCHC salinity and temperature data at Station 94.
 - b. Final ACCESS table: "SALTEMPBC"
 - c. Export to: "SALTEMPBC.xls"
 - i. Remove header row (Row 1)
 - ii. Set format to 1 decimal
 - iii. Set column width to 8
 - d. Save as: "SALTEMPBC.prn"
 - e. Run executable "SALTEMPBC.exe"
 - f. Creates formatted file for putting into "run_data": "SALTEMPBC"

Step 1-8: Input the river inflows and temperatures for data group G1

1. Define the number of river inflows

2. Define the grids, the connecting grids (I,J), and the vertical distribution of the flows for the river inflows
3. Input the time series data for flow, temperature, and salinity for each defined grid as block data using the ACCESS database "RIVERFLOW.accdb"
 - a. This database contains seven primary data tables:
 - i. "Flows_1999" and "Flows_2000_2009" contain daily INTB flows by reach, including all reaches used to develop port flows
 - ii. "MR_TCB_flows" contains monthly flows for Manatee River and Terra Ceia Bay, outside the INTB domain, from TBEP data
 - iii. "Lake_Tarpon_flows" contains daily observed flows for Lake Tarpon
 - iv. "HR_TBC_flows" contains daily observed flows for Hillsborough River Dam and Tampa Bypass Canal S-160
 - v. "Stream_Temp" contains daily stream temperatures based on monthly observed data for each inflow location
 - b. Final ACCESS table: "RIVERFLOW"
 - c. Export to: "RIVERFLOWS.xls"
 - i. Remove header row (Row 1)
 - ii. Set format to 5 decimals
 - iii. Set column width to 11
 - d. Save as: "RIVERFLOW.prn"
 - e. Run executable: "RIVERFLOW.exe"
 - f. Create formatted file for putting into "run_data": "RIVERFLOW"

Step 1-9: Input the point source discharges for data group G2

1. Define the number of point source discharges
2. Define the grids and the distribution of the flows over the vertical
3. Input the time series data for each point source flow, temperature, and salinity by defined grid as block data using the ACCESS database "DIRECTPS.accdb"
 - a. This database contains one primary data table, "Direct_PS_flows", which includes monthly discharges directly to Tampa Bay from Howard F. Curren and the City of Clearwater North and Northeast facilities.
 - b. Final Access table: "DIRECTPS"
 - c. Export to: "DIRECTPS.xls"
 - i. remove header row (Row 1),
 - ii. set format to 5 decimals,
 - iii. set column width to 11 in Excel file
 - d. Save as : "DIRECTPS.prn"
 - e. Run executable: "DIRECTPS.exe"
 - f. Creates formatted file for putting into "run_data": "DIRECTPS"

Step 1-10: Input the groundwater inflows for data group G3

1. Define the number of zones in the groundwater.map
2. Input the time series data for each zone including flow, temperature, and salinity using the ACCESS database "GROUNDWATER.accdb"
 - a. This database contains one primary data table, "GW_flows", which includes the time-invariant inflows to 5 bay regions.
 - b. Final Access table: "GW"
 - c. Export to: "GW.xls"
 - i. remove header row (Row 1),
 - ii. remove column B
 - iii. set format to 6 decimals,

- iv. set column width to 10 in Excel file
- d. Convert to: "GW.prn"
- e. Run executable: "GW.exe"
- f. Creates formatted file for putting into "run_data": "GW"

Step 1-11: Input the meteorology data for data group H

1. Define the type of met data inputs and basic meteorologic parameters
2. Input the time series data for each zone including wind speed, wind direction, short-wave radiation, air temperature, relative humidity, barometric pressure, cloud cover, light extinction coefficient, precipitation, evaporation using the ACCESS database "MET.accdb"
 - a. This database contains four primary data tables:
 - i. "DoverET" contains daily ET rates (in/day)
 - ii. "Rainfall" contains daily segment-specific rainfall (in/day) derived from TBEP rainfall estimates
 - iii. "OTB_Secchi" contains monthly OTB average Secchi disk depth (m) from EPCHC data
 - iv. "SP CLW MET DATA" contains hourly St. Petersburg/Clearwater Airport wind speed, wind direction, air temperature, atmospheric pressure, cloud cover, Dover solar radiation, and Dover relative humidity
 - b. Final Access table: "MET"
 - c. Export to: "MET.xls"
 - i. Remove header row (Row 1)
 - ii. Set format to 5 decimals
 - iii. Set column width to 11
 - d. Save as: "MET.prn"
 - e. Run executable: "MET.exe"
 - f. Creates formatted file for putting into "run_data": "MET"

model_grid

The OTB model_grid file has three primary sections, these are;

- Specification of the Sigma Levels
- Horizontal Grid Data
- Specification of the Locations of Thin Wall Dams

Appendix B presents detailed descriptions of the variables within the model_grid file and the input formats.

groundwater.map

The OTB groundwater.map file defines what grid cells are located within 5 different zones within the model grid domain. A total of 5 zones are defined, and the groundwater.map file simply provides for the I,J grid numbers and what zone (1 to 5) that cell resides in. Within the run_data file, in Section G under the Diffuser-GW the constant flow values, for each of the 5 segments, are read in at time 0.0 and the same value for time 99999.0 which sets these values to the constants for the full simulation.

STEP 2: RUNNING THE MODEL

Prior to running the OTB Hydrodynamic Model, it is recommended that the user set up a base directory for the runs, an example might be as follows:

```
OTBhydro>baserun
```

Under the baserun sub-directory set up two directories, model_input, model_output. Put the three model inputs (run_data, model_grid, and groundwater.map), the model executable (ecom3d33.exe), and the batch file (discussed below) in the model_input directory. Subsequent runs could then be identified by other names (i.e. baserun_1) and set up under their own sub-directory.

A batch file <file>.bat is set up in the same directory as the executable and the model input files. The batch file has the structure as shown below.

```
set ncpus=4
set OMP_NUM_THREADS=4
ecom3d33.exe
pause
```

The first two lines of code in the batch file are common methods for setting the number of threads on the computer being utilized. The specification of the number of threads is critically important to model run time.

A command line window will then appear. The window will provide a count on the number of days the model has run. JHIST=<number> provides the count by days that the model has run. The number of processors in the computer running the model its processor speed and other factors will define how long the model will run.

STEP 3: MODEL POST PROCESSING FOR CALIBRATION GRAPHICS AND STATISTICS

Figure 2 provides a flow chart showing the output file structure from ECOMSED and what each will be utilized for. ECOMSED outputs 4 primary files for use in post-processing data. These are;

- GCMTSR
- GCM_TRAN
- GCM_GEOM
- GCMPLT
- Wave.out

The GCMTSR file provides high temporal resolution time series results at specified grid cells based upon the control variables defined in Data Group C. In data group C specific cells for the GCMTSR output are specified for elevations as a group; currents, temperature and salinity as a group; and fluxes as a group. The GCM_TRAN and GCM_GEOM files provide the hydrodynamic input to the RCA water quality model. The GCMPLT file provides output for the full grid for all variables at specified intervals. The Wave.out file contains the time series of

information from the wave model. For the purposes of the calibration manual, only the GCMTSR and Wave.out results are discussed.

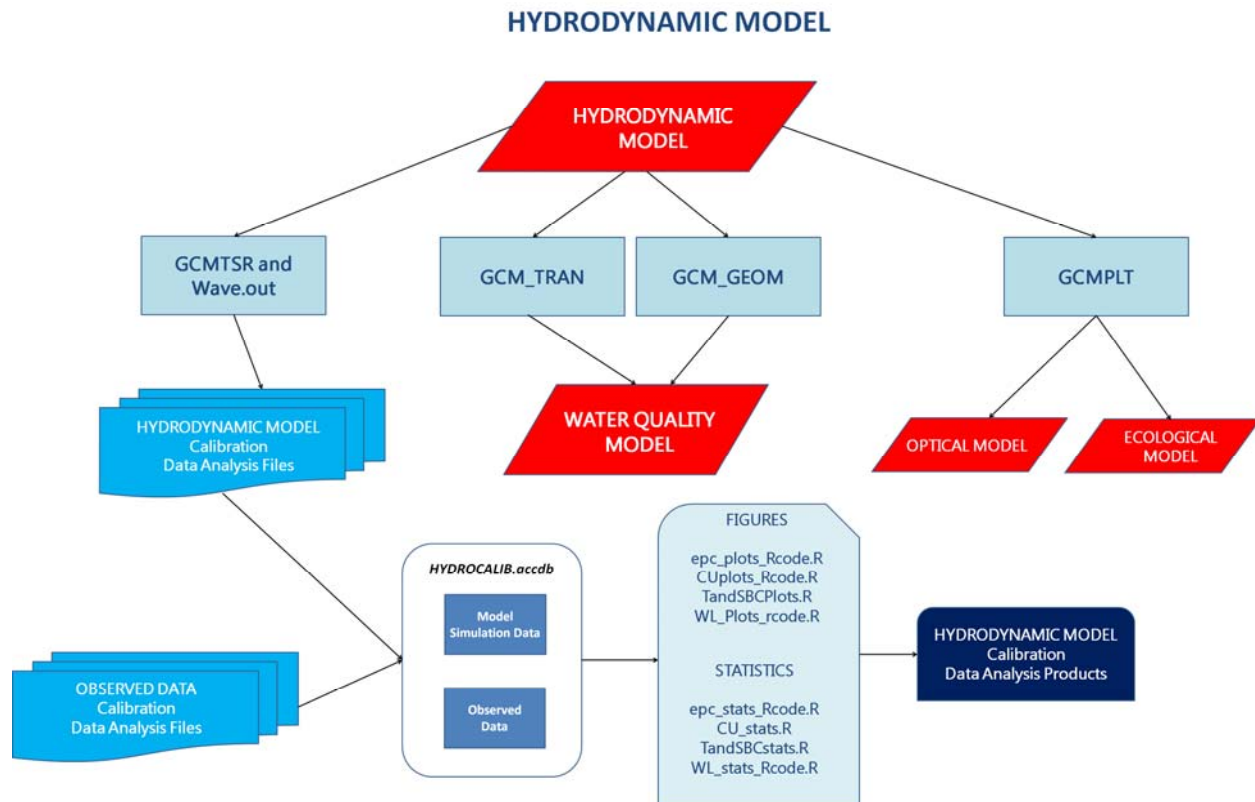


Figure 2: Hydrodynamic Output File Structure

Post-Processing GCMTSR to create .txt time series files

A program has been provided with the model run files called *tsrdat.exe*. This program reads the GCMTSR binary file and creates a series of .txt files. The .txt files each then correspond to associated data files that are used for comparison of the model results to the data files. The .txt data files and the .txt output files from the model base run are provided on the hard drive included with the presentation.

Tables 1 through 4 below show the one-to-one correspondence of the .txt files from the model with the .txt files from the data. When the *tsrdat.exe* program is run two files are created, these are *HeaderData.txt* and *tsrhdr.txt*. These files provide the structure and order of the data within the post-processed model files.

Post-Processing Wave.out to create .txt time series files

A program has been provided with the model run files called *Waveread.exe*. This program reads the Wave.out file and creates a series of .txt files for the various output stations identified. The .txt files each then correspond to time series output from the wave component of the hydrodynamic model. The specific variables output in order following the time and I,J locations include;

- Wave Height
- Wave Period

- Wave Direction
- Bottom Shear Stress

As there is not direct data for comparison of the wave model results, only the model output files are discussed herein.

Table 1 – Correspondence of data .txt files with model .txt files for water surface elevation

<u>Water Level</u>	
<i>DATA</i>	<i>Model Output</i>
OPTs.txt	elvpts.txt
MKBs.txt	elvpts.txt
SPs.txt	elvpts.txt
PMs.txt	elvpts.txt
tarponcanal.txt	elvpts.txt

Table 2 – Correspondence of data .txt files with model .txt files for currents

<u>Currents</u>	
<i>DATA</i>	<i>Model Output</i>
OPT_CU_TS.txt	velpt2.txt
WI_CU_3TS.txt	velpt3.txt
SS_CU_TS.txt	velpt1.txt

Table 3 – Correspondence of data .txt files with model .txt files for salinity and temperature

Salinity and Temperature	
<i>DATA</i>	<i>Model Output</i>
epc14.txt	velpt27.txt
epc23.txt	velpt37.txt
epc16.txt	velpt28.txt
epc33.txt	velpt32.txt
epc42.txt	velpt11.txt
epc47.txt	velpt12.txt
epc60.txt	velpt16.txt
epc62.txt	velpt17.txt
epc70.txt	velpt56.txt
epc44.txt	velpt53.txt
epc64.txt	velpt19.txt
epc8.txt	velpt52.txt
epc6.txt	velpt50.txt
epc7.txt	velpt51.txt
epc9.txt	velpt24.txt
epc11.txt	velpt25.txt
epc13.txt	velpt26.txt
epc19.txt	velpt29.txt
epc21.txt	velpt36.txt
epc24.txt	velpt38.txt
epc25.txt	velpt39.txt
epc28.txt	velpt30.txt
epc32.txt	velpt31.txt
epc36.txt	velpt7.txt
epc38.txt	velpt8.txt
epc40.txt	velpt9.txt
epc41.txt	velpt10.txt
epc50.txt	velpt14.txt
epc51.txt	velpt15.txt
epc52.txt	velpt54.txt
epc55.txt	velpt55.txt
epc61.txt	velpt13.txt
epc63.txt	velpt18.txt
epc65.txt	velpt20.txt
epc66.txt	velpt21.txt
epc67.txt	velpt22.txt
epc68.txt	velpt23.txt
epc71.txt	velpt57.txt
epc73.txt	velpt58.txt
epc80.txt	velpt60.txt
epc81.txt	velpt33.txt
epc82.txt	velpt34.txt
epc84.txt	velpt35.txt
epc90.txt	velpt40.txt
epc91.txt	velpt41.txt
epc92.txt	velpt42.txt
epc93.txt	velpt43.txt
epc94.txt	velpt44.txt
epc95.txt	velpt45.txt
epc96.txt	velpt46.txt

Table 4 – Correspondence of data .txt files with model .txt files for flux through CCC

<u>Flux</u>	
<i>DATA</i>	<i>Model Output</i>
East032312.txt	flxpts.txt
West032312.txt	flxpts.txt
East041712.txt	flxpts.txt
West041712.txt	flxpts.txt

R-CODE FOR POST PROCESSING

Code for recreating hydrodynamic calibration figures in R

R Manual for Hydrodynamic Model Calibration:

Objective: Create hydrodynamic calibration figures and statistics using an open resource data analysis package. Below is an example of how to import data, get data ready for use, and create figures using ggplot2 in R.

General information: R stores functions as libraries. We will be using the library ggplot2 for all of the graphics. The first time you use ggplot2 you must download the library: 1) select the cran mirror (choose USA(CA2), and then select install package(S),choose ggplot2, all of this is under the Packages drop down menu, afterwards you then need to tell R to open the library by using the following command. `library(ggplot2)`

Data Import:

The primary functions to read data into R are `read.table` (for text files) and `read.csv` (for csv files). To read in a data file 1) name the file (if you do not name the file then R will open up the data in the workspace), 2) tell R what function to use (`read.table` or `read.csv`), 3) tell R where the file is, 4) how the data are separated, 5) and if there are column names (header).

Note: R is case sensitive

A skeleton example

```
x<-read.table("C:/tmpR/example.txt",sep="",header=FALSE)
```

Since the file was named the file is now stored in R

```
read.table("C:/tmpR/example.txt",sep="",header=FALSE)
```

	V1	V2
1	1	4
2	2	6
3	5	8
4	10	10

The above command was not named, therefore, the file will appear in the workspace

Note: R is case sensitive

Below is the code used to reproduce the EPC 47 surface salinity figure (Figure 3-44) from the hydrodynamic calibration report

Data files used: model output file = *velpt12.txt*, observed data file = *epc47.txt*

Both files are text files, separated with spaces, and no header

Code to read the model output file velpt12.txt, this will take a few minutes due to the size of the file

```
velpt12<-read.table("C:/Users/john/Documents/Calibration Report  
FINAL/velpt12.txt",sep="",header=FALSE)
```

Can use command head to verify the file was read in correctly `head(velpt12)`

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15
1	0.00208	6	1.59	0	0	21.11	13.76	0	0	21.11	13.76	0	0	21.11	13.76
2	0.00625	6	1.91	0	0	25.33	16.51	0	0	25.33	16.51	0	0	25.33	16.51
3	0.01042	6	1.91	0	0	25.33	16.51	0	0	25.33	16.51	0	0	25.33	16.51
4	0.01458	6	1.91	0	0	25.33	16.51	0	0	25.33	16.51	0	0	25.33	16.51
5	0.01875	6	1.91	0	0	25.33	16.51	0	0	25.33	16.51	0	0	25.33	16.51
6	0.02292	6	1.91	0	0	25.33	16.51	0	0	25.33	16.51	0	0	25.33	16.51

	V16	V17	V18	V19	V20	V21	V22	V23
1	0	0	21.11	13.76	0	0	21.11	13.76
2	0	0	25.33	16.51	0	0	25.33	16.51
3	0	0	25.33	16.51	0	0	25.33	16.51
4	0	0	25.33	16.51	0	0	25.33	16.51
5	0	0	25.33	16.51	0	0	25.33	16.51
6	0	0	25.33	16.51	0	0	25.33	16.51

Can use the command `dim` to verify all rows and columns were read correctly `dim(velpt12)`

[1] 963600 23

Code to read the observed data file *epc47.txt*

`epc47<-read.table("C:/Users/john/Documents/Calibration Report FINAL/epc47.txt",sep=" ",header=FALSE)`

Verify the data were read in correctly `head(epc47)`

	V1	V2	V3	V4	V5	V6	V7
1	18.6	18.7	19.1	26.0	26.0	25.8	8843
2	13.9	13.7	13.8	27.3	25.8	25.6	9515
3	20.9	21.4	22.2	27.5	27.3	27.0	10211
4	24.2	24.2	24.2	28.5	28.5	28.5	11027
5	24.1	24.2	24.3	30.4	30.3	30.2	11699
6	29.0	29.1	29.1	32.0	32.0	31.9	12539

`dim(epc47)` [1]

120 7

Add a header to each file
Create a vector of the names, for example: `names<-c("a","b")`
Vector of names for velpt12

`namesvelpt<-c("Time","k","dz","u1","v2","s1","t1","u2","v2","s2","t2","u3","v3","s3","t3","u4",
"v4","s4","u5","v5","s5","t5")`

Use the function `colnames` to add the vector of column headers created above `colnames(velpt12)<-namesvelpt`

Verify by using the head command `head(velpt12)`

	Time	k	dz	u1	v2	s1	t1	u2	v2	s2	t2	u3	v3	s3	t3	u4
1	0.00208	6	1.59	0	0	21.11	13.76	0	0	21.11	13.76	0	0	21.11	13.76	0
2	0.00625	6	1.91	0	0	25.33	16.51	0	0	25.33	16.51	0	0	25.33	16.51	0
3	0.01042	6	1.91	0	0	25.33	16.51	0	0	25.33	16.51	0	0	25.33	16.51	0
4	0.01458	6	1.91	0	0	25.33	16.51	0	0	25.33	16.51	0	0	25.33	16.51	0
5	0.01875	6	1.91	0	0	25.33	16.51	0	0	25.33	16.51	0	0	25.33	16.51	0
6	0.02292	6	1.91	0	0	25.33	16.51	0	0	25.33	16.51	0	0	25.33	16.51	0
	v4	s4	t4	u5	v5	s5	t5									
1	0	21.11	13.76	0	0	21.11	13.76									
2	0	25.33	16.51	0	0	25.33	16.51									
3	0	25.33	16.51	0	0	25.33	16.51									
4	0	25.33	16.51	0	0	25.33	16.51									
5	0	25.33	16.51	0	0	25.33	16.51									
6	0	25.33	16.51	0	0	25.33	16.51									

Repeat the above steps to add column names to the epc47 dataset `namesepc<-c("TB","TM","TS","SB","SM","SS","Year")`

key: TB=temperature bottom, TM=temperature middle, TS = temperature surface, SB=salinity bottom, SM = salinity middle, SS = salinity surface, Year = time variable

```
colnames(epc47)<-namesepc
head(epc47)
```

	TB	TM	TS	SB	SM	SS	Year
1	18.6	18.7	19.1	26.0	26.0	25.8	8843
2	13.9	13.7	13.8	27.3	25.8	25.6	9515
3	20.9	21.4	22.2	27.5	27.3	27.0	10211
4	24.2	24.2	24.2	28.5	28.5	28.5	11027
5	24.1	24.2	24.3	30.4	30.3	30.2	11699
6	29.0	29.1	29.1	32.0	32.0	31.9	12539

For plotting purposes need to standardize the time variable in both datasets
Create new vector that includes the time variable from velptxx + 730121 and add it to the velpt12 dataset

```
velpt12$Time_stand<-(velpt12$Time+ 730121)
Use command head to verify column was added head(velpt12)
```

	Time k		dz u1 v2		s1		t1 u2 v2		s2		t2 u3 v3		s3		t3 u4	
1	0.00208	6	1.59	0	0	21.11	13.76	0	0	21.11	13.76	0	0	21.11	13.76	0
2	0.00625	6	1.91	0	0	25.33	16.51	0	0	25.33	16.51	0	0	25.33	16.51	0
3	0.01042	6	1.91	0	0	25.33	16.51	0	0	25.33	16.51	0	0	25.33	16.51	0
4	0.01458	6	1.91	0	0	25.33	16.51	0	0	25.33	16.51	0	0	25.33	16.51	0
5	0.01875	6	1.91	0	0	25.33	16.51	0	0	25.33	16.51	0	0	25.33	16.51	0
6	0.02292	6	1.91	0	0	25.33	16.51	0	0	25.33	16.51	0	0	25.33	16.51	0
	v4	s4	t4	u5	v5	s5	t5	Time	stand							
1	0	21.11	13.76	0	0	21.11	13.76		730121							
2	0	25.33	16.51	0	0	25.33	16.51		730121							
3	0	25.33	16.51	0	0	25.33	16.51		730121							
4	0	25.33	16.51	0	0	25.33	16.51		730121							
5	0	25.33	16.51	0	0	25.33	16.51		730121							
6	0	25.33	16.51	0	0	25.33	16.51		730121							

```
Repeat the steps for the epcxx data epc47$Time_stand<-
((epc47$Year/24)+730121)
head(epc47)
```

	TB	TM	TS	SB	SM	SS	Year	Time	stand
1	18.6	18.7	19.1	26.0	26.0	25.8	8843		730489
2	13.9	13.7	13.8	27.3	25.8	25.6	9515		730517
3	20.9	21.4	22.2	27.5	27.3	27.0	10211		730546
4	24.2	24.2	24.2	28.5	28.5	28.5	11027		730580
5	24.1	24.2	24.3	30.4	30.3	30.2	11699		730608
6	29.0	29.1	29.1	32.0	32.0	31.9	12539		730643

The velpt12 dataset included data from 1999 so need to subset the data to only included data from 2000 to 2009, use the function `subset` to do this

```
velpt12_date_2000<-subset(velpt12,Time_stand>=730486)
```

You now have a new dataset called `velpt12_data_2000` that was created from `velpt12` and included all data from 2000 to 2009

The data are ready to create the figure from the hydrodynamic calibration report

Note: verify you opened up the library `ggplot2` from the above steps if not do it now `library(ggplot2)`

Skeleton of the `ggplot2` function,

`name.plot <- ggplot(name of data set, aes(xaxis=what column, yaxis=what column)) + geom_line` (or `geom_point` etc., this command specifies what kind of plot wanted) + `scale_x_continuous(breaks=c())` use this command if you want specific breaks in the x axis, `continuous_scale(aesthetics = c("x", "xmin", "xmax", "xend", "xintercept"), scale_name = "position_c", palette = identity, breaks = ..1, expand = expand, guide = "none")`

Naming the plot allows you to add additional layers

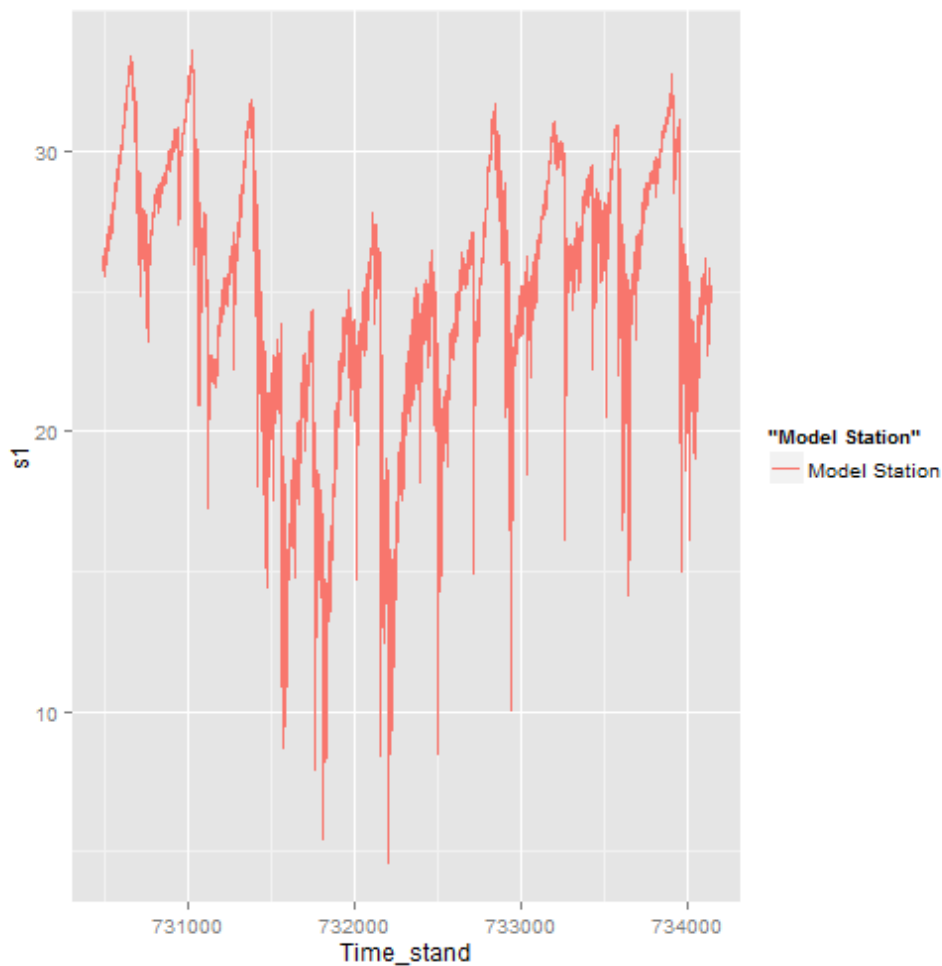
Note: the plot will not appear in the R window when named unless you specifically tell R to open it

First layer, `velpt12_data_2000` as a line

```
velpt12SS <- ggplot(velpt12_data_2000, aes(x=Time_stand, y=s1, colour="Model Station")) +  
  geom_line(shape="line")
```

See plot in R

```
velpt12SS
```

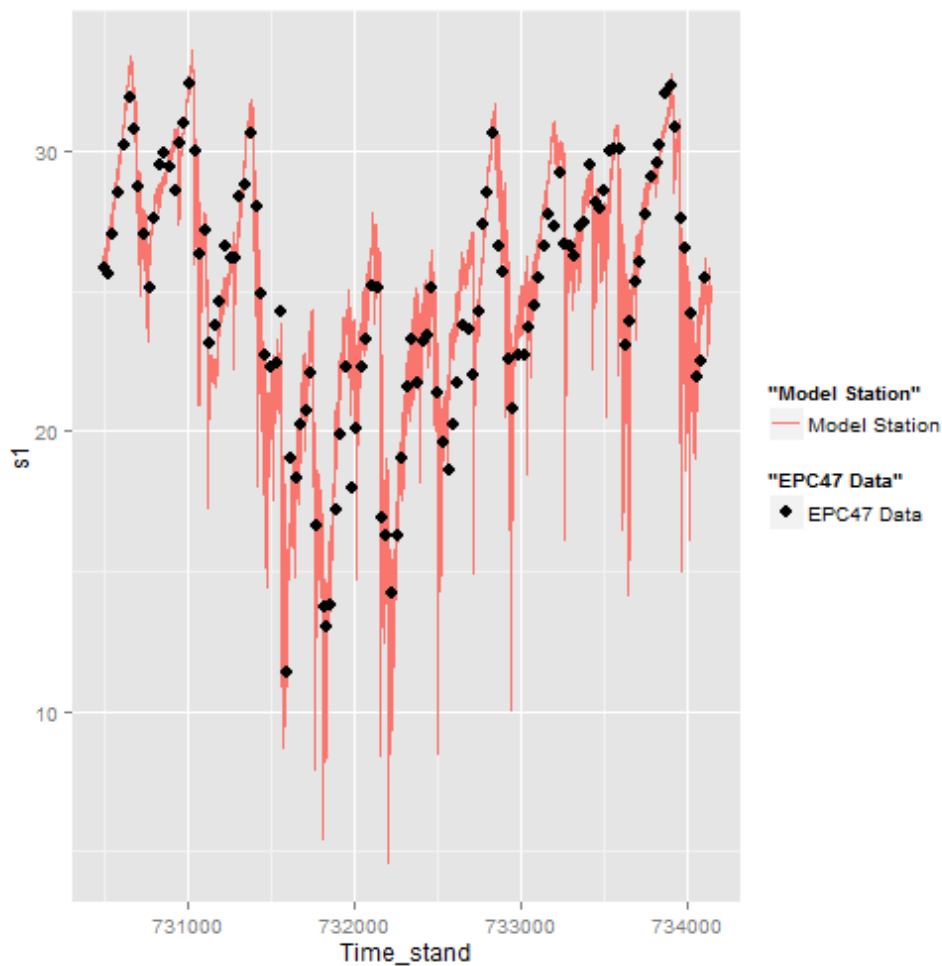


Add the observed epc47 data as points he observed EPC data as a point plot (note: when adding a second layer, need to specify data outside of the aes line), shape, size, and color

```
epc47_SS<-velpt12SS + geom_point(aes(x=Time_stand,y=SS, shape="EPC47
Data"),data=epc47,size=3,colour="black")
```

The above plot now as both layers, velpt12SS and epc47_SS

```
epc47_SS
```

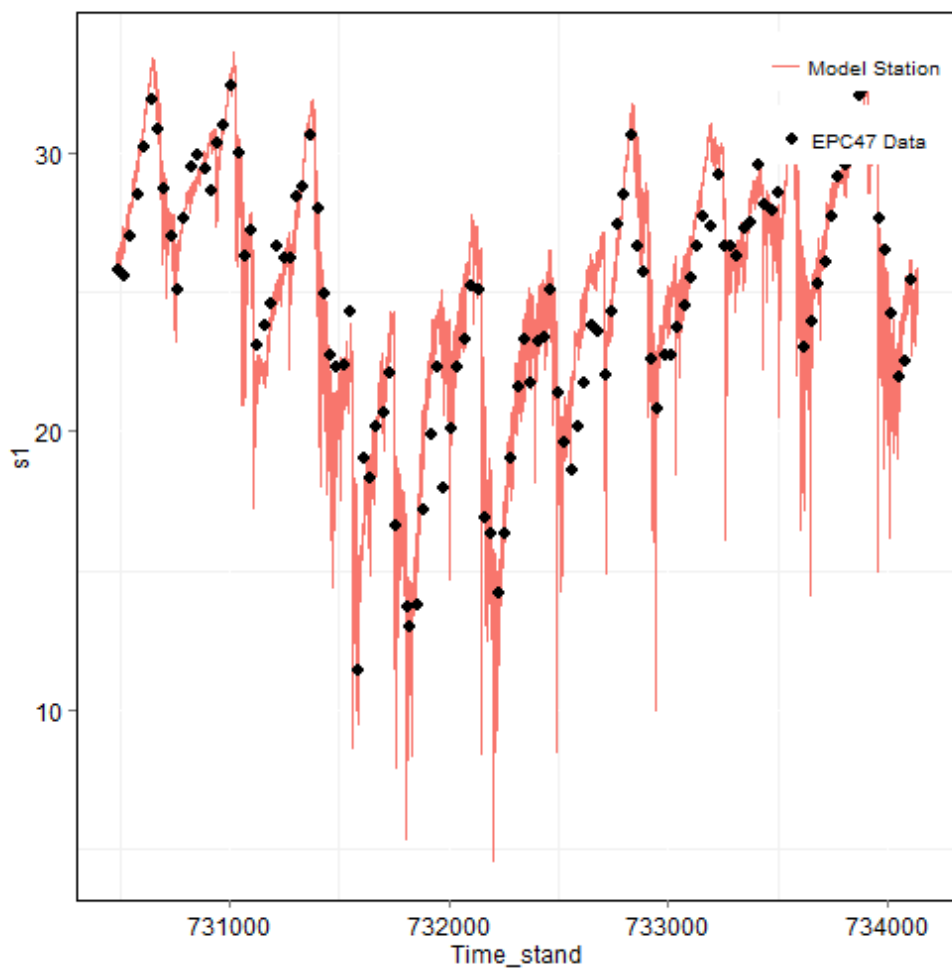


Specify the theme for the plot, this theme removes the default gray background and adds the location of the legend

```
epc47_SS_theme <- epc47_SS + theme(legend.title = element_blank()) + theme(legend.justification = c(1, 1), legend.position = c(1, 1)) +
  theme(panel.border = element_rect(colour = "black", fill = NA)) +
  theme(axis.text = element_text(colour = "black", size = 12)) +
  theme(panel.background = element_rect(fill = NA, colour = "black")) +
  theme(legend.key = element_rect(fill = NA))
```

Can see the difference

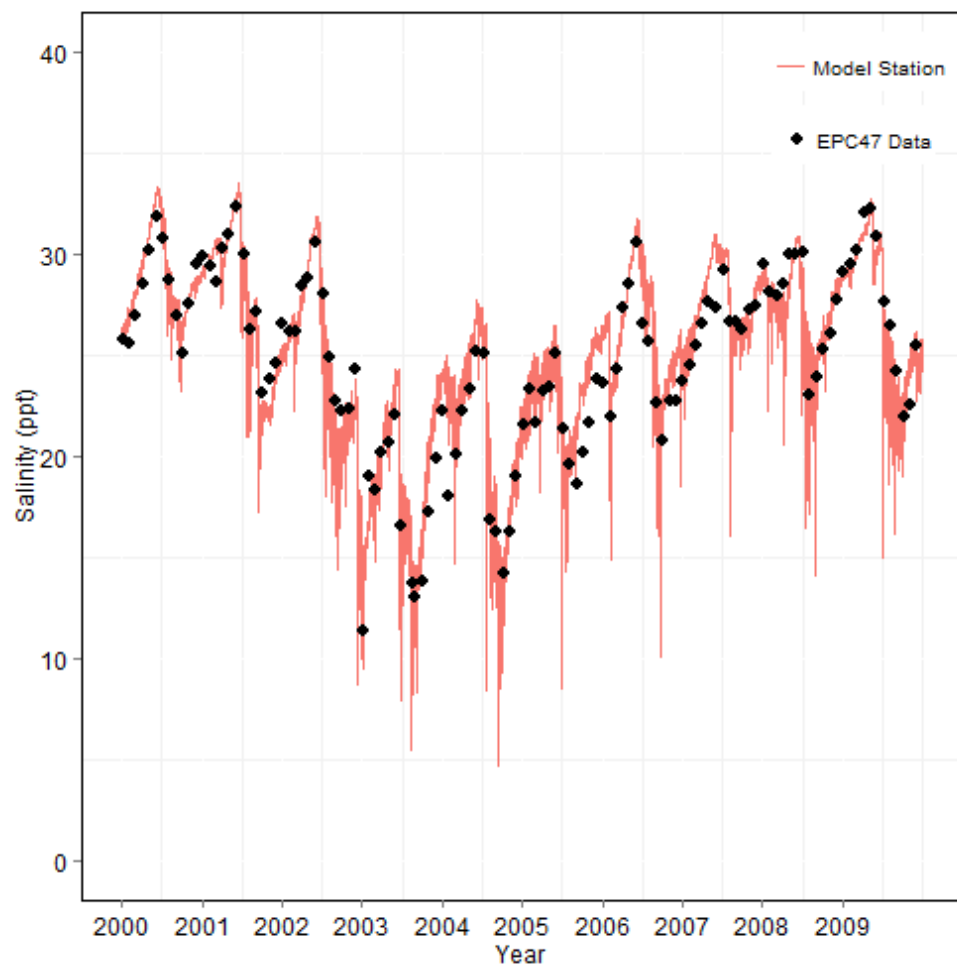
`epc47_SS_theme`



Add the x and y labels

```
epc47_SS_axis<-epc47_SS_theme+ scale_x_continuous(breaks=c(730486.00,730852.00, 731217.00,
731582.00, 731947.00, 732313.00, 732678.00, 733043.00,
733408.00,733774.00),labels=c(2000,2001,2002,2003,2004,2005,2006,2007,2008,2009))+
  ylim(0,40)+ xlab("Year") +ylab("Salinity (ppt)")
```

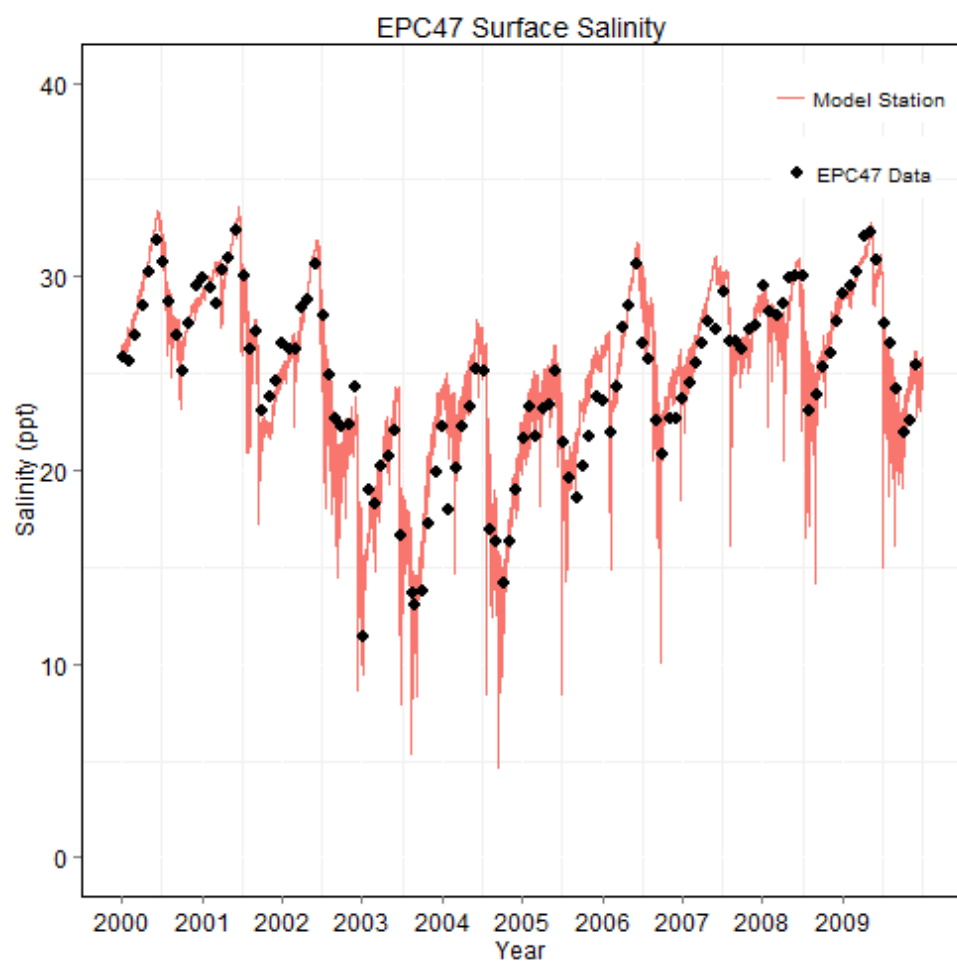
What the figure now looks like `epc47_SS_axis`



Add title

```
epc47_SS_title<-epc47_SS_axis+ ggtitle("EPC47 Surface Salinity")
```

```
epc47_SS_title
```

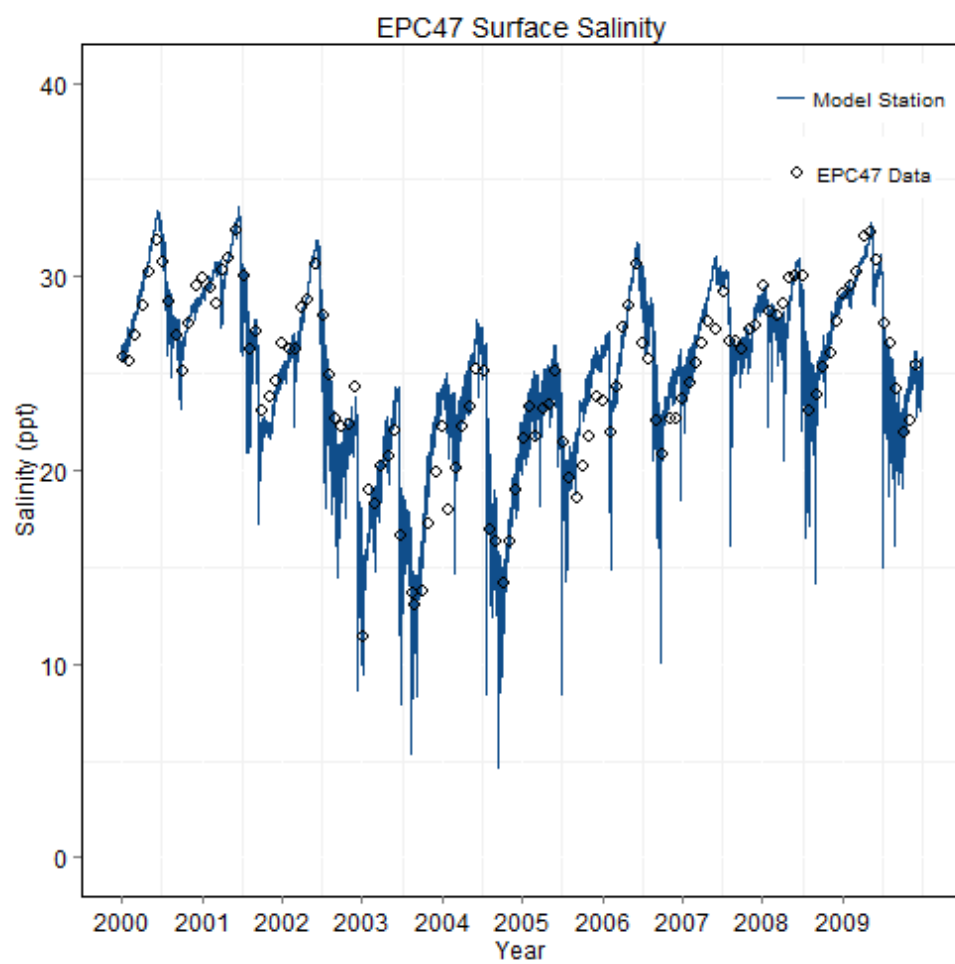


Change the color of the lines and style of the circles

```
epc47_SS_final<-epc47_SS_title+ scale_colour_manual(values=c("dodgerblue4"))+  
  scale_shape_manual(values=1)
```

The figure is now ready

```
epc47_SS_final
```



Save the plot as a pdf on the computer using the function `ggsave`, base code (name of figure, where you want it saved and the name you want the graph to be called, size)

```
ggsave(epc47_SS_final, filename="C:/Users/john/Documents/oldtampa  
bay/EPC_Plots/EPC47SurfaceSalinity.pdf", height=7.5, width=7.5)
```

Code for recreating hydrodynamic calibration figures in R

```
Old Tampa Bay Project
Hydrodynamic Output, R code for statistics

EPC statistics, sample EPC47, data used: model output=velpt12.txt, observed data = epc47.txt
See manual on R figures for details about libraries and reading and formatting data in R
See files: epc_stats_Rcode.R, CU_stats.R, TandSBCstats.R, WL_stats_Rcode.R for code for all statistics in the
hydrodynamic calibration report

velpt12<-read.table("C:/Users/john/Documents/Calibration Report FINAL/velpt12.txt", sep="", header=FALSE)
epc47<-read.table("C:/Users/john/Documents/Calibration Report FINAL/epc47.txt", sep = "", header=FALSE)

Add colnames
namesvelpt<-c("Time", "k", "dz", "u1", "v2", "s1", "t1", "u2", "v2", "s2", "t2", "u3", "v3", "s3", "t3", "u4", "v4", "s4", "t4",
              "u5", "v5", "s5", "t5")
colnames(velpt12)<-namesvelpt

namesepc<-c("TB", "TM", "TS", "SB", "SM", "SS", "Year")
colnames(epc47)<-namesepc

Standardize epc date
epc47$Time_stand<-(epc47$Year/24)

Interpolation surface salinity
Need to install library signal and then code in R for the library to open

library(signal)

Loading required package: MASS

Attaching package: 'signal'

The following objects are masked from 'package:stats':

filter, poly

Interpolate salinity values at match time stamp values to the observed data using the function interp1
EPC47_SS<-data.frame(actual=interp1(velpt12$Time, velpt12$s1, epc47$Time_stand, 'linear'))

Create a new data set with the interpolated salinity values and observed salinity values
EPC47_SS2 <- data.frame(interp=EPC47_SS[,1], actual=epc47$SS)

Statistics to compare between salinity values
Linear model between the interpolated and observed (actual) values

EPC47_SS.lm <- lm(interp~actual, data=EPC47_SS2);summary(EPC47_SS.lm )

Call:
lm(formula = interp ~ actual, data = EPC47_SS2)

Residuals:
    Min       1Q   Median       3Q      Max
-2.674  -0.720  -0.091   0.858   3.315

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.1965     0.6363    5.02 1.8e-06 ***
actual        0.8891     0.0253   35.20 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.25 on 118 degrees of freedom
Multiple R-squared:  0.913, Adjusted R-squared:  0.912
F-statistic: 1.24e+03 on 1 and 118 DF, p-value: <2e-16

Calculate error between the interpolated and observed (actual) values
EPC47_SS.error <- EPC47_SS2$actual-EPC47_SS2$interp

Functions to calculate RMSE,MAE and ME

RMSE <- function(EPC47_SS.error){
  sqrt(mean(EPC47_SS.error^2))
}

mae <- function(EPC47_SS.error){
```

```
mean(abs(EPC47_SS.error))
}

me <- function(EPC47_SS.error){
  mean(EPC47_SS2$actual-EPC47_SS2$interp)
}

Use the above functions to calculate the statistics

RMSE_EPC47_SS <- RMSE(EPC47_SS.error)

Round value

RMSE_EPC47_SS_round<- lapply(RMSE_EPC47_SS, round, 1);RMSE_EPC47_SS_round
```

```
[[1]]
[1] 1.4
```

```
MAE_EPC47_SS <- mae(EPC47_SS.error)
MAE_EPC47_SS_round<- lapply(MAE_EPC47_SS, round, 1);MAE_EPC47_SS_round
```

```
[[1]]
[1] 1.1
```

```
ME_EPC47_SS <- me(EPC47_SS.error)
ME_EPC47_SS_round<- lapply(ME_EPC47_SS, round, 1);ME_EPC47_SS_round
```

```
[[1]]
[1] -0.4
```

```
R2_EPC47_SS<- summary(EPC47_SS.lm)$r.squared
R2_EPC47_SS_round<- lapply(R2_EPC47_SS, round, 2);R2_EPC47_SS_round
```

```
[[1]]
[1] 0.91
```

```
N <- length(EPC47_SS2[,1]);N
```

```
[1] 120
```

```
Repeat above code for bottom salinity, change to appropriate columns
EPC47_SB<-data.frame(actual=interp1(velpt12$Time,velpt12$s5,epc47$Time_stand,'linear'))
EPC47_SB2 <- data.frame(interp=EPC47_SB[,1], actual=epc47$SB)

Linear model
EPC47_SB.lm <- lm(interp~actual, data=EPC47_SB2);summary(EPC47_SB.lm )
```

```
Call:
lm(formula = interp ~ actual, data = EPC47_SB2)

Residuals:
    Min       1Q   Median       3Q      Max
-2.594  -0.721  -0.126   0.974   2.787

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    2.4199     0.6198     3.9 0.00016 ***
actual         0.9066     0.0241    37.6 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.14 on 118 degrees of freedom
Multiple R-squared:  0.923, Adjusted R-squared:  0.922
F-statistic: 1.41e+03 on 1 and 118 DF, p-value: <2e-16
```

```
Error
EPC47_SB.error <- EPC47_SB2$actual-EPC47_SB2$interp

RMSE,MAE and ME functions

RMSE <- function(EPC47_SB.error){
  sqrt(mean(EPC47_SB.error^2))
}

mae <- function(EPC47_SB.error){
```

```
mean(abs(EPC47_SB.error))
}
me <- function(EPC47_SB.error){
  mean(EPC47_SB2$actual-EPC47_SB2$interp)
}

Statistics
RMSE_EPC47_SB <- RMSE(EPC47_SB.error)
RMSE_EPC47_SB_round<- lapply(RMSE_EPC47_SB, round, 1);RMSE_EPC47_SB_round
```

```
[[1]]
[1] 1.2
```

```
MAE_EPC47_SB <- mae(EPC47_SB.error)
MAE_EPC47_SB_round<- lapply(MAE_EPC47_SB, round, 1);MAE_EPC47_SB_round
```

```
[[1]]
[1] 1
```

```
ME_EPC47_SB <- me(EPC47_SB.error)
ME_EPC47_SB_round<- lapply(ME_EPC47_SB, round, 1);ME_EPC47_SB_round
```

```
[[1]]
[1] -0.1
```

```
R2_EPC47_SB<- summary(EPC47_SB.lm)$r.squared
R2_EPC47_SB_round<- lapply(R2_EPC47_SB, round, 2);R2_EPC47_SB_round
```

```
[[1]]
[1] 0.92
```

```
N <- length(EPC47_SB2[,1]);N
```

```
[1] 120
```

```
Repeat above code for surface temperature, change to appropriate columns
EPC47_TS<-data.frame(actual=interp1(velpt12$Time,velpt12$t1,epc47$Time_stand,'linear'))
EPC47_TS2 <- data.frame(interp=EPC47_TS[,1], actual=epc47$TS)

Linear model
EPC47_TS.lm <- lm(interp~actual, data=EPC47_TS2);summary(EPC47_TS.lm )
```

```
Call:
lm(formula = interp ~ actual, data = EPC47_TS2)

Residuals:
    Min       1Q   Median       3Q      Max
-3.928 -0.519  0.076  0.592  2.070

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.2719     0.3725   -0.73    0.47
actual        0.9621     0.0153   63.00 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.912 on 118 degrees of freedom
Multiple R-squared:  0.971, Adjusted R-squared:  0.971
F-statistic: 3.97e+03 on 1 and 118 DF, p-value: <2e-16
```

```
Error
EPC47_TS.error <- EPC47_TS2$actual-EPC47_TS2$interp
RMSE,MAE and ME functions

RMSE <- function(EPC47_TS.error){
  sqrt(mean(EPC47_TS.error^2))
}

mae <- function(EPC47_TS.error){
  mean(abs(EPC47_TS.error))
}
me <- function(EPC47_TS.error){
  mean(EPC47_TS2$actual-EPC47_TS2$interp)
}
```

```
Statistics
RMSE_EPC47_TS <- RMSE(EPC47_TS.error)
RMSE_EPC47_TS_round<- lapply(RMSE_EPC47_TS, round, 1);RMSE_EPC47_TS_round
```

```
[[1]]
[1] 1.5
```

```
MAE_EPC47_TS <- mae(EPC47_TS.error)
MAE_EPC47_TS_round<- lapply(MAE_EPC47_TS, round, 1);MAE_EPC47_TS_round
```

```
[[1]]
[1] 1.3
```

```
ME_EPC47_TS <- me(EPC47_TS.error)
ME_EPC47_TS_round<- lapply(ME_EPC47_TS, round, 1);ME_EPC47_TS_round
```

```
[[1]]
[1] 1.2
```

```
R2_EPC47_TS<- summary(EPC47_TS.lm)$r.squared
R2_EPC47_TS_round<- lapply(R2_EPC47_TS, round, 2);R2_EPC47_TS_round
```

```
[[1]]
[1] 0.97
```

```
N <- length(EPC47_TS2[,1]);N
```

```
[1] 120
```

```
Repeat above code for bottom temperature, change to appropriate columns
EPC47_TB<-data.frame(actual=interp1(velpt12$Time, velpt12$t5, epc47$Time_stand, 'linear'))
EPC47_TB2 <- data.frame(interp=EPC47_TB[,1], actual=epc47$TB)
```

```
Linear model
EPC47_TB.lm <- lm(interp~actual, data=EPC47_TB2);summary(EPC47_TB.lm )
```

```
Call:
lm(formula = interp ~ actual, data = EPC47_TB2)

Residuals:
    Min       1Q   Median       3Q      Max
-4.692  -0.464   0.007   0.662   2.236

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.1897     0.4271   -0.44    0.66
actual         0.9722     0.0176   55.16 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.04 on 118 degrees of freedom
Multiple R-squared:  0.963, Adjusted R-squared:  0.962
F-statistic: 3.04e+03 on 1 and 118 DF, p-value: <2e-16
```

```
Error
EPC47_TB.error <- EPC47_TB2$actual-EPC47_TB2$interp
RMSE,MAE and ME functions

RMSE <- function(EPC47_TB.error){
  sqrt(mean(EPC47_TB.error^2))
}

mae <- function(EPC47_TB.error){
  mean(abs(EPC47_TB.error))
}

me <- function(EPC47_TB.error){
  mean(EPC47_TB2$actual-EPC47_TB2$interp)
}

Statistics
RMSE_EPC47_TB <- RMSE(EPC47_TB.error)
RMSE_EPC47_TB_round<- lapply(RMSE_EPC47_TB, round, 1);RMSE_EPC47_TB_round
```

```
[[1]]
[1] 1.3
```

```
MAE_EPC47_TB <- mae(EPC47_TB.error)
MAE_EPC47_TB_round<- lapply(MAE_EPC47_TB, round, 1);MAE_EPC47_TB_round
```

```
[[1]]
[1] 1.1
```

```
ME_EPC47_TB <- me(EPC47_TS.error)
ME_EPC47_TB_round<- lapply(ME_EPC47_TB, round, 1);ME_EPC47_TB_round
```

```
[[1]]
[1] 0.8
```

```
R2_EPC47_TB<- summary(EPC47_TB.lm)$r.squared
R2_EPC47_TB_round<- lapply(R2_EPC47_TB, round, 2);R2_EPC47_TB_round
```

```
[[1]]
[1] 0.96
```

```
N <- length(EPC47_TB2[,1]);N
```

```
[1] 120
```

```
Output statistics as a .csv file

Combine RMSE values using rbind function
EPC_47_RMSE<-rbind(RMSE_EPC47_SS_round, RMSE_EPC47_SB_round, RMSE_EPC47_TS_round, RMSE_EPC47_TB_round, deparse.level=0)

EPC_47_RMSE
```

```
      [,1]
[1,] 1.4
[2,] 1.2
[3,] 1.5
[4,] 1.3
```

```
Combine MAE values using rbind function
EPC_47_MAE<-rbind(MAE_EPC47_SS_round, MAE_EPC47_SB_round, MAE_EPC47_TS_round, MAE_EPC47_TB_round, deparse.level=0)

EPC_47_MAE
```

```
      [,1]
[1,] 1.1
[2,] 1
[3,] 1.3
[4,] 1.1
```

```
Combine ME values using rbind function
EPC_47_ME<-rbind(ME_EPC47_SS_round, ME_EPC47_SB_round, ME_EPC47_TS_round, ME_EPC47_TB_round, deparse.level=0)

EPC_47_ME
```

```
      [,1]
[1,] -0.4
[2,] -0.1
[3,] 1.2
[4,] 0.8
```

```
Combine R2 values using rbind function
EPC_47_R2<-rbind(R2_EPC47_SS_round, R2_EPC47_SB_round, R2_EPC47_TS_round, R2_EPC47_TB_round, deparse.level=0)

EPC_47_R2
```

```
      [,1]
[1,] 0.91
[2,] 0.92
[3,] 0.97
[4,] 0.96
```

```
Create vector for row names
Parameters<-c("Salinity Surface", "Salinity Bottom", "Temperature Surface", "Temperature Bottom")

Bind all values together using function cbind
EPC47_STATS<-cbind(Parameters,EPC_47_RMSE,EPC_47_MAE,EPC_47_ME,EPC_47_R2)

Write column names using function colnames
names<-c("Paramters","RMSE","MAE","ME","R2")
colnames(EPC47_STATS)<-names

EPC47_STATS
```

	Paramters	RMSE	MAE	ME	R2
[1,]	"Salinity Surface"	1.4	1.1	-0.4	0.91
[2,]	"Salinity Bottom"	1.2	1	-0.1	0.92
[3,]	"Temperature Surface"	1.5	1.3	1.2	0.97
[4,]	"Temperature Bottom"	1.3	1.1	0.8	0.96

```
Export file as an .csv file
write.csv(EPC47_STATS,"C:/Users/john/Documents/old tampa bay/EPC_stats_output/HB_TEMP_STATS.csv", row.names=FALSE)

Calculations on current data
Sunshine Skyway (NOAA T02010)

velpt1<- read.table("C:/Users/john/Documents/Calibration Report FINAL/velpt1.txt", sep=" ", header=FALSE)
namesvelpt<-c("Time", "k", "dz", "u1", "v1", "s1", "t1", "u2", "v2", "s2", "t2", "u3", "v3", "s3", "t3", "u4", "v4", "s4",
"t4", "u5", "v5", "s5", "t5"

colnames(velpt1)<-namesvelpt

Square the columns
v <- data.frame(lapply(velpt1[c(4:5)], function(z)z^2))

Sum across the row
v$Sum <- rowSums(v)
square root the sum
v$sqrtsum <- v$Sum^.5

Repeat for the other two columns
v2 <- data.frame(lapply(velpt1[c(8:9)], function(z)z^2))
v$Sum2 <- rowSums(v2)
v$sqrtsum2 <- v$Sum2^.5

Add the calculated square root values together
v$Sum3<-v$sqrtsum+v$sqrtsum2

Divide the sume by 2
v$divide<-v$Sum3/2

Multiply by 100
v$magn<-v$divide*100

Make a new data set with the calculated magnitude
v_time<-data.frame(velpt1$Time,v$magn)
v_names<-c("Time", "Magnitude")
colnames(v_time)<-v_names

head(v_time)
```

	Time	Magnitude
1	0.00208	0
2	0.00625	0
3	0.01042	0
4	0.01458	0
5	0.01875	0
6	0.02292	0

APPENDIX A

RUN_DATA FILE DOCUMENTATION

DATA GROUP A: Computational and Output Characteristics

Comment

COM = user specified comment for run information

Comment - Run Options

COM = header for run options

A1 - Run Options

HYDTYPE	=	"INTERNAL" - use internal (ECOM) hydrodynamics
	=	"EXTERNAL" - use external hydrodynamics input from 'hqi_tran' file
WAVEDYN	=	"NEGLECT" - no effect of surface waves on bottom friction
	=	"SMBMODEL" - include effects of waves on bottom friction, internal calculation of waves using SMB theory
	=	"DONMODEL" - include effects of waves on bottom friction, internal calculation of waves using Donelan Theory
	=	"EXTERNAL" - include effects of waves on bottom friction, wave parameters input from 'wave_input' file, (external calculation using WAM or HISWA)
TRACER	=	"INCLUDE" - dissolved tracer transport will be simulated
	=	"NEGLECT" - no simulation of dissolved tracer transport
SEDTRAN	=	"INCLUDE" - sediment transport will be simulated
	=	"NEGLECT" - no simulation of sediment transport
CHEMTRAN	=	"INCLUDE" - sediment-bound tracer transport will be simulated
	=	"NEGLECT" - no simulation of sediment-bound transport
SEDTYPE	=	"BOTH" - cohesive and non-cohesive sediment transport
	=	"MUD" - cohesive sediment transport only
	=	"SAND" - non-cohesive sediment transport only
PARTICLE	=	"INCLUDE" - particle tracking will be simulated
	=	"NEGLECT" - no simulation of particle tracking

Comment Run Computational Characteristics

COM = header to identify run computational characteristics

A2 - Run Computational Characteristics

DTI	=	time step in seconds of the internal mode (the maximum allowable time step in seconds can be found in "gcmpert")
SPLIT	=	number of time steps between the internal and external modes. Please refer to Section 8.1 for more details.
IRAMP	=	number of time steps over which all model forcing functions are ramped from zero to their full values linearly
IYR, IMO, IDA, IHR	=	year, month, day, hour of model start time and IYR should be a 4-digit number
NHYD	=	number of time steps between each hydrodynamic transport field input from "hqi_tran", only used if HYDTYPE = "EXTERNAL"
SGW	=	semi-prognostic coefficient based on Sheng, Greatbatch and Wang (2001). Please see Section 8-4 for details. If SGW = 1 and TOR = "PROGNOSTIC," the model will be in full prognostic mode (Default = 1.0)
WETEPS	=	Drying Depth
WETMIN	=	Minimum Depth for Wetting

Comment Run Output Characteristics

COM = header for Run Output characteristics

A3 - Run Output Characteristics

NSTEPS	=	number of time steps in the model run
IPRINT	=	print interval in number of time steps
IPRTSTART	=	time in number of time steps at which printing will begin
RESTART	=	"COLD START" _ all initial conditions are set to zero
	=	"HOT START" _ all initial conditions are input from file "restart"
TOR	=	"BAROTROPIC" _ 2_D calculation (bottom stress calculated in ADVAVE)
	=	"PROGNOSTIC" _ 3_D calculation (bottom stress calculated in PROFU & PROFV)
	=	"DIAGNOSTIC" _ 3_D calculation with temperature and salinity held fixed
	=	"TEMP-ONLY" - 3-D prognostic run while initial salinity values are held throughout the simulation
	=	"SALT-ONLY" - 3-D prognostic run while initial temperature fields are held throughout the simulation
ADVECT	=	"LINEAR" _ no momentum advection terms
	=	"NON_LINEAR" _ include momentum advection terms

SCHEME	=	"CENTRAL" - central finite difference scheme for advection
	=	"UPWIND" - upwind finite difference scheme for advection
	=	"SMOLAR_R" - finite difference scheme due to Smolarkiewicz and using reclusive formulation for antidiffusive velocities
	=	"SMOLAR_2" - finite difference scheme due to Smolarkiewicz and using two passes for corrections of numerical diffusion

Comment - Run Print Characteristics

COM	=	header to identify Run Print Characteristics
-----	---	--

A4 - Run Print Characteristics

DEV	=	primary output device for viewing "gcmpmt"
	=	"SCR" _ for 15 columns across the page, suitable for printing on a screen with no wrap around
	=	"LPR" _ for 25 columns across the page, suitable for printing on a laser or line printer
VSX	=	vertical slice in the x ($\square$ 1) direction of various model quantities included in the "gcmpmt" file
	=	"Y" _ vertical slices
	=	"N" _ no vertical slices
JROW	=	j number at which the vertical slice in the x ($\square$ 1) direction will be taken
	=	0, for VSX="N"
VSX	=	vertical slice in the y ($\square$ 2) direction of various model quantities included in the "gcmpmt" file
	=	"Y" _ vertical slices
	=	"N" _ no vertical slices
ICOL	=	I number at which the vertical slice in the y ($\square$ 2) direction will be taken
	=	0, for VSX="N"
PTU	=	U velocity included in "gcmpmt"
	=	"Y" _ include
	=	"N" _ omit
PTV	=	V velocity included in "gcmpmt"
	=	"Y" _ include
	=	"N" _ omit
PTW	=	W velocity included in "gcmpmt"
	=	"Y" _ include
	=	"N" _ omit
PTAM	=	horizontal mixing included in "gcmpmt"
	=	"Y" _ include
	=	"N" _ omit
PTS	=	salinity and conservative tracer included in "gcmpmt"

	=	"Y" _ include
	=	"N" _ omit
PTT	=	temperature included in "gcmprt"
	=	"Y" _ include
	=	"N" _ omit
PRHO	=	density included in "gcmprt"
	=	"Y" _ include
	=	"N" _ omit
PTQ2	=	turbulent kinetic energy included in "gcmprt" for closure vertical mixing
	=	"Y" _ include
	=	"N" _ omit
PTL	=	mixing length included in "gcmprt" for closure vertical mixing
	=	"Y" _ include
	=	"N" _ omit
PTKM	=	mixing KM included in "gcmprt" for closure vertical mixing
	=	"Y" _ include
	=	"N" _ omit
PTKH	=	mixing KH included in "gcmprt" for closure vertical mixing
	=	"Y" _ include
	=	"N" _ omit

DATA GROUP B: Hydrodynamic Characteristics

Comment - Constants of the Model Problem

COM = header for Constants of the Model Problem

B1 - Constants of the Model Problem

BFRIC = minimum bottom friction coefficient (non_dimensional)
If user wants to specify spatially variable (2D) bottom friction coefficients, specify BFRIC as "VARI" (6X, A4) and provide "bfri2d.inp" file. Please refer to Table 10-26 for more details of format specification of "bfri2d.inp" file.

ZOB = bottom roughness coefficient in meters

NU = coefficient in time filter (non_dimensional)
= 0.1 (recommended value)

THETA = weighting factor (0-1); 0 - explicit and 1- implicit scheme
= 0.225 (recommended value)

ALPHA = advection time scale for temperature and salinity at the boundary time "HOURS" over which the boundary values reach their full specified value during the flood cycle from the values exiting at the end of the ebb cycle. *Caution: If the user does not want boundary relaxation (∇-folding), specify ALPHA = 0.

TLAG = friction time scale (Hours) for barotropic radiation boundary condition (only needed if user selects BCTYPE as PCLAMP)

NWAVE = number of time steps between each update of bottom friction coefficient, only used if WAVEDYN is not "NEGLECT". This defines the interval (in time steps) that the wave model is called.

BCTYPE = barotropic radiation boundary condition types
"CLAMPED" - clamp boundary condition (no radiation)
"PCLAMP" - partially clamped. Note: if user selects PCLAMP type B.C., user must provide TLAG in Hours
"OCLAMP" - optimized clamp which is same as RANDB except 8t is non unity.
"RANDB" - Reid and Bodine type boundary condition
"IRANDB" - inverted Reid and Bodine type boundary condition)(8₉ = 1)

Comment - Horizontal Mixing Characteristics

COM = header for horizontal mixing characteristics

B2 - Horizontal Mixing Characteristics

HORZMIX	=	"CONSTANT" _ value given for HORCON is scaled in each grid element relative to the smallest grid element
	=	"CLOSURE" _ value given for HORCON is used in Smagorinsky's formula for mixing
HORCON	=	value used as a constant or in Smagorinsky's formula based on HORZMIX (non_dimensional)
		If user wants to specify spatially variable (2D) HORCON, specify HORCON as "VARI" (6X,A4) and provide "horcon2d.inp" file. Please refer to Table 10-25 for more details of format specification of "horcon2d.inp" file.
HPRNU	=	horizontal Prandtl number _ ratio of horizontal viscosity to horizontal diffusivity (momentum mixing/dispersive mixing)
	=	1.0 (recommended value)

Comment - Vertical Mixing Characteristics

COM = header for vertical mixing characteristics

B3 - Vertical Mixing Characteristics

VERTMIX	=	"CONSTANT" _ value given for UMOL applies everywhere
	=	"CLOSURE" _ value given to UMOL is background mixing
	=	"EMPIRICAL" - empirical vertical turbulent mixing based on Kent and Pritchard (1959) and Officer (1976)
UMOL	=	constant or background mixing in m ² /sec
	=	1.0E_06 (recommended value if VERTMIX = "CLOSURE") or "EMPIRICAL".
		If user wants to specify spatially variable (2D) UMOL, specify UMOL as "VARI" (6X,A4) and provide "umol2d.inp" file. Please refer to table 10-27 for more details of format specification of "umol2d.inp" file
VPRNU	=	vertical Prandtl number _ ratio of vertical viscosity to vertical diffusivity (momentum mixing/diffusive mixing) for VERTMIX = "CONSTANT"
	=	If VERTMIX = "EMPIRICAL", VPRNU acts as coefficient ∇ in the empirical mixing equation (8-9a)

DATA GROUP C: Result Evaluation

Comment - Computational History Output for Plotting

COM = user specified comment

C1 - Number and Averaging Interval of Computational History Output Sets

JHM = number of times all information necessary for plotting will be written in "gcmlpt" and "part_location" (if PARTICLE = "INCLUDE")

AVGE = interval in number of time steps or hours for averaging the elevations, temperature, salinity and currents for all grid elements

IPLTFORM = "USER" or "(left in blank spaces)" user specifies the "gcmlpt" output intervals.
"AUTO" output interval will be generated by ECOMSED based on the values of PLTZERO, AVGE and JHM

PLTZERO = The beginning timestep or hour of "gcmlpt" output (TZERO)

OPTAVG = indicating the units of AVGE and PLTZERO.
"HOURL": time control is in hour,
"NDT": time control is in number of timesteps.
If the OPTAVG is in hour and AVGE is not an integer multiple of timestep ECOMSED will stop and ask for another AVGE period.

C2 - Averaging Interval for Skill Assessment (GCMTSR File)

a. **Comment**

COM = user specified comment

b. **Averaging Interval**

ISKILL = interval in number of time steps for averaging the elevations, temperature, salinity and currents and cross sectional fluxes for user specified grid elements
= 0, no element stored in "gcmtsr" for skill assessment

NOTE: If ISKILL=0, then go to Data Group C6
(Computational Results for Water Quality Model)

C3 - Computed Time Series for Elevations

a. Comment

COM= user specified comment

b. Number of Grid Elements

EPTS = number of grid elements for which time series of elevations are to be stored in "gcmts"

NOTE: If EPTS=0, then go to Data Group C4 (Computed Time Series for Currents, Temperature and Salinity)

c. Location of Grid Elements

INXIE = I number of grid element

NXJE = J number of grid element

C4 - Computed Time Series for Currents, Temperature, Salinity and Transport Quantities

a. Comment

COM = user specified comment for computed time series for temperature, salinity and currents

b. Number of Grid Elements

VPPTS = number of grid elements for which time series of temperature, salinity and currents are to be stored in "gcmts"

NOTE: If VPPTS=0, then go to Data Group C5 (Computed Time Series for Cross Sectional Fluxes)

c. Location of Grid Elements

INXIV = I number of grid element

INXJV = J number of grid element

C5 - Computed Time Series for Cross Sectional Fluxes

a. Comment

COM = user specified comment for computed time series for cross sectional fluxes

b. Number of Cross Sections

FPTS = number of cross sections for which time series of fluxes are to be stored in "gcmtr"

NOTE: If FPTS=0, then go to Data Group C.6 (Computation Results for Water Quality Model)

c. Location of Cross Sections

ISFLX = I grid element in which cross section begins

JSFLX = J grid element in which cross section begins

DIRFLX = direction of cross_section

= "IDIR" - I direction

= "JDIR" - J direction

NFLXE = number of grid elements

C6 - Computation Results for Water Quality Model

a. Comment

COM = user specified comment

b. Number and Averaging Interval of Computation Result Output Sets

JTM = number of times all information necessary for the water quality model input is generated

NAVE = interval in number of time steps for averaging the elevations and currents to be used as input in the water quality model

ITRNFORM = 0 : user specified time steps for writing the output
1: ECOM will generate the writing block (i.e. section 6.c.)

IZERO = # of time steps to skip before start to writing 'gcm_tran' information. IZERO should not be '0' if 'COLD START'. If ITRNFORM = 0, IZERO will be ignored.

IWET = 0 : entire grid output
1 : wet grid only output

NOTE: If JTM = 0, then go to Data Group D (Standard Level Declaration)

c. Time in Number of Time Steps for Writing the Output

IF ITRANSFORM = 0

IF ITRANSFORM = 1 skip this block

IETRAC = time in number of time steps at which the information will be output, necessary for water quality model input

NOTE: IETRAC relative to start of run (independent of RESTART option specified)

DATA GROUP D: Standard Level Declaration

Comment

COM = user specified comment about the standard levels

D1 - Number of Standard Levels

IKSL = number of standard levels (< 50)

NOTE: To reduce the amount of computer storage required, keep the number of standard levels (IKSL) at a minimum, i.e., $IKSL < 50$.

D2 - Standard Levels

DPTHSL = depth of standard level in meters with respect to surface water level

NOTE: It is not necessary to include the surface level, although it may be included. If not, constituent values associated with the first level below the surface will be applied to the depth between the surface and the first standard level. Extrapolation to the surface may cause incorrect representation of the vertical profile. To ensure proper interpolation of data from standard level to sigma level, each bottom_most sigma level must be bracketed by two standard levels. These standard levels must contain data.

DATA GROUP E: Initial Temperature and Salinity

Comment

COM = user specified comment for temperature and salinity data

E1 - Initial Temperature and Salinity Data Option

OPT*TSI = "FIXED" _ initial temperature and salinity data are constant for each standard level.
 = "DATA" _ initial temperature and salinity data vary horizontally and vertically _ data read in from data file "init_tands".
 If OPT*TSI = "DATA", then go to Data Group F (Open Boundary Condition Information)

E2 - Initial Temperature Data

TSI = temperature in °C for each standard layer

E3 - Initial Salinity Data

SSI = salinity in psu for each standard layer

DATA GROUP F: Open Boundary Condition Information

F1 - Elevation Boundary Conditions

a. Comment

COM = user specified comment

b. Number of Grid Elements and Option

NUMEBC = total number of elevation boundary grid elements.
If NUMEBC = 0, then go to Data Group G
(Discharge Information)

OPTEBC = "DATA" _ use Elevation Boundary Conditions
OPTION 1 _ Time Variable Data
= "TIDAL CONSTITUENT" _ use Elevation
Boundary Conditions OPTION 2 _ Computer
Generated Data from Tidal Constituents

c. Elevation Boundary Conditions

OPTION 1 TIME VARIABLE DATA

i. Location of Grid Elements

IETA = I number of grid element where elevation is
specified

JETA = j number of grid element where elevation is
specified

ICON = I number of connecting grid element (nearest
interior non_ boundary grid element)

JCON = j number of connecting grid element (nearest
interior non_ boundary grid element)

NOTE: Every boundary element should have a connecting
interior grid element.

ii. Time of Observation

TIME = time in hours
= 0.0 for initial time

NOTE: TIME is absolute time measured from beginning of
COLD START run and incremented with each
subsequent HOT START run.

iii. Elevation Data

EBDRY = boundary elevation data in meters at time "TIME"

NOTE: Sequence (TIME/EBDRY) repeated for each observation. Final "TIME" must be greater than $(NSTEPS \times DTI)/3600$ (the duration of the run), for COLD START runs, and greater than $IEND + (NSTEPS \times DTI)/3600$ for HOT START runs.

F2 - Time Variable Temperature and Salinity Boundary Conditions

a. Comment

COM = user specified comment

b. Temperature and Salinity Boundary Conditions

i. Time of Observation

TIME = time in hours
= 0.0 for initial time

NOTE: TIME is absolute time measured from beginning of COLD START run and incremented with each subsequent HOT START run.

c. Location of Grid Elements, Temperature and Salinity Data

ITAS = i number of grid element where temperature and salinity are specified
JTAS = j number of grid element where temperature and salinity are specified
TBDRYSL = temperature in °C at time "TIME" for each standard level (not sigma level)
SBDRYSL = salinity in psu at time "TIME" for each standard level (not sigma level)

DATA GROUP G: Discharge Information

G1 - Time Variable Tributary Inflow

a. **Comment**

COM = user specified comment for discharge

b. **Number of Grid Elements**

NUMQBC = total number of discharge boundary grid elements.
If NUMQBC = 0, then go to Data Group G.2
(Time Variable Offshore Intake/Outfall (Diffuser)
Discharges)

c. **Location of Grid Elements, Vertical Distribution of flow**

IQD = i number of grid element discharge enters/leaves
JQD = j number of grid element discharge enters/leaves
IQC = i number of connecting exterior boundary grid
element
JQC = j number of connecting exterior boundary grid
element
VQDIST = percentage (not fraction) of total discharge QDIS
apportioned to each model layer from surface to
bottom at location (IQD,JQD)

d. **Time of Observation**

TIME = time in hours
= 0.0 for initial time

e. **Discharge Data**

QDIS = discharge flow in m³/sec
> 0.0 (positive) for flow into the model domain
(river/outfall)
< 0.0 (negative) for flow out of the model domain
(intake)

f. **Temperature Data**

TDIS = temperature of discharge in °C

g. **Salinity Data**

SDIS = salinity of discharge in psu

G2 - Time Variable Offshore Intake/Outfall (Diffuser) Discharges

a. Comment

COM = user specified comment for discharge

b. Number of Grid Elements

NUMDBC 1 = total number of discharge grid elements. If NUMDBC = 0, then go to Data Group G3 (Time Variable Offshore Intake/Outfall (Diffuser) Discharges in Loops)

c. Location of Grid Elements/Vertical Distribution of Intake/Outfall Diffuser Discharge

IDD = i number of grid element diffuser enters/leaves
JDD = j number of grid element diffuser enters/leaves
VDDIST = percentage (not fraction) of total discharge DQDIFF apportioned each model layer from surface to bottom at location (IDD,JDD)

d. Time of Observation

TIME = time in hours
= 0.0 for initial time

e. Discharge Data

QDIFF = diffuser discharge in m³/sec

f. Temperature Data

TDIFF = temperature of diffuser discharge in °C

f. Salinity Data

SDIFF = salinity of diffuser discharge in psu

G3 - Time Variable Offshore Intake/Outfall (Diffuser) Discharges in Loops

a. Comment

COM = user specified comment for discharge in loops

b. Number of Grid Elements

NUMDBC2 = total number of discharge grid elements. If NUMDBC2 = 0, then go to Data Group G4 (Groundwater Data).

c. Location of Grid Elements/Vertical Distribution of Intake/Outfall Diffuser Discharge

IDD = i number of grid element diffuser enters/leaves

JDD = j number of grid element diffuser enters/leaves

VDDIST = percentage (not fraction) of total discharge DQDIFF apportioned each model layer from surface to bottom at location (IDD,JDD)

KBM1 = KB-1

d. Time of Observation

TIME = time in hours
= 0.0 for initial time

e. Discharge Data

DQDIFF = diffuser discharge in m³/sec. Even though DQDIFF(N) and DQDIFF(N+1) is a coupling intake/outfall pair, the DQDIFF(N) and DQDIFF(N+1) need not be equal.

f. Temperature Data

DTDIFF = temperature increase/decrease of diffuser discharge in °C. Only effective on diffuser having positive values of DQDIFF. This increase/decrease of temperature is with respect to the temperature of corresponding intake.

g. Salinity Data

DSDIFF = salinity increase/decrease of diffuser discharge in psu. Only effective on diffuser having positive values of DQDIFF.

DATA GROUP H: Meteorological Data

1. Comment

COM = user specified comment for meteorological data

2. Meteorological Data Option

OPTMBC = “AVERAGED” _ a single value for each meteorological parameter is used for all grid element at each time _ use Meteorological Data OPTION 1 _ Time Variable Surface Heat Flux Data, Salt Flux Data and Wind Data

= “SYNOPTIC” _ spatially varying meteorological parameter values are specified for every grid element at each time _ use Meteorological Data OPTION 2 _ Averaged Time Variable Surface Heat Flux Data and Salt Flux Data and Synoptic Time Variable Wind and Atmospheric Pressure Data

= “AANDBFLX” - heat flux sub-model based on the work of Ahsan and Blumberg (1999). The local heat flux is determined from local surface temperature - use Meteorological Data OPTION 3 - Time Variable Surface Heat Flux Parameters, Salt Flux Data and Wind Data

= “LANDPFLX” - heat flux sub-model based on the work of Large and Pond (1982). The local heat flux is determined from local surface temperature - use Meteorological Data OPTION 3 - Time Variable Surface Heat Flux Parameters, Salt Flux Data and Wind Data.

= “RANDMFLX” - heat flux sub-model based on the work of Rosati and Miyakoda (1988). The local heat flux is determined from local surface temperature - use Meteorological Data OPTION 3 - Time Variable Surface Heat Flux Parameters, Salt Flux Data and Wind Data.

= “SYNOPLNP”, “SYNOPLNP”, “SYNOPLNP”: These options allow the user to specify spatially varying wind data along with heat flux sub-model options “AANDBFLX”, “LANDPFLX”, and “RANDMFLX”, respectively.

ALAT = Latitude (in degrees, median of modeling domain)

ALON = Longitude (in degrees, median of modeling domain) (0 ~ -180 for western hemisphere)

TR = Fraction of short wave radiation absorbed in surface layer

WNSH = Wind sheltering coefficient (default = 1.0)

REFL = Reflection of shortwave radiation at surface (0.0-1.0) (0.0 - nothing reflected; 1.0 - complete reflection) 0.1 (recommended value)

OPTEXTC = Option for extinction coefficients. If “VARI”, spatially varying extraction coefficients specified in “extc2d.inp” will be used. If OPTEXTC is left in blank or other than “VARI”, the model will use time-varying but spatially uniform EXTC specified in OPTION 3 below.

3. Meteorological Data

OPTION 3 - Time Variable Surface Heat Flux Parameters, Salt Flux Data and Wind Data

i. Time of Observation

TIME = time in hours
= 0.0 for initial time

ii. Met Data Input

WDS = wind speed in m/sec
WDD = direction of wind in degrees from which the wind blows, measured clockwise from north
SWOBS = observed shortwave radiation in watts/m²
AIRTMP = air temperature in °C
RELHUM = relative humidity in percent
BAROP = barometric pressure in mbar
CLD = cloud cover fraction (0.0 to 1.0)
EXTC = extinction coefficient
QPREC = amount of precipitation in m/year
QEVAP = amount of evaporation in m/year

APPENDIX B

MODEL_GRID FILE

DOCUMENTATION

Data Group A: Comment for Grid Information

COM = user specified comment for grid information

Data Group B: Vertical Segmentation

1. Comment

2. Number of Sigma Levels

IKB = number of sigma levels

3. Sigma Levels

Z = depth of the interface between sigma levels
 $-1.0 < Z < 0.0$

NOTE: Total number = IKB.

Data Group C: Horizontal Segmentation

1. Comment

COM = user specified comment for horizontal segmentation

2. I Index and J Index

IIX = index in the $\square 1$ direction

IJY = index in the $\square 2$ direction

3. Grid Information

I = I number of grid element in the $\square 1$ direction

J = j number of grid element in the $\square 2$ direction

H1 = distance in meters in the $\square 1$ direction at center of grid element

H2 = distance in meters in the $\square 2$ direction at center of grid element

H = average depth of grid element in meters (at mean water level)
= MLW + tidal amplitude

ANG = angle in degrees between east and $\square 1$ direction measured in a counter-clockwise direction

YGRID = latitude of grid center in degrees (positive for northern hemisphere) to compute the Coriolis parameter

XGRID = longitude of grid center in degrees (Note: model does not use this. Only used for postprocessing purposes)

DATUM = datum of grid element in meters (above some reference elevation)

NOTE: Total number of wet grid $\square$ total number of grid elements. Grid information need be specified for wet points only and it is not necessary to specify grid information for other grid elements. H must be sufficiently large in order to remain wet at low tide.

Data Group D: Specification of Thin Dams

1. Comment

COM = user specified comment for thin dams.

2. Number of Thin Dams

NUMTDAM = Number of thin dams

NOTE: If there is no thin dam specified in the model domain, user can either specify NUMTDAM = 0 or skip Data Group D.

3. Location of Thin Dams

ISTDAM(N) = I number of grid element in which thin dam begins

JSTDAM(N) = j number of grid element in which thin dam begins

DIRTDAM(N) = direction of thin dam (cross-section)

“IDIR” - cross-section is in the $>_1$ direction

“JDIR” - cross-section is in the $>_2$ direction

NTDAM(N) = number of grid elements in the thin dam cross-section